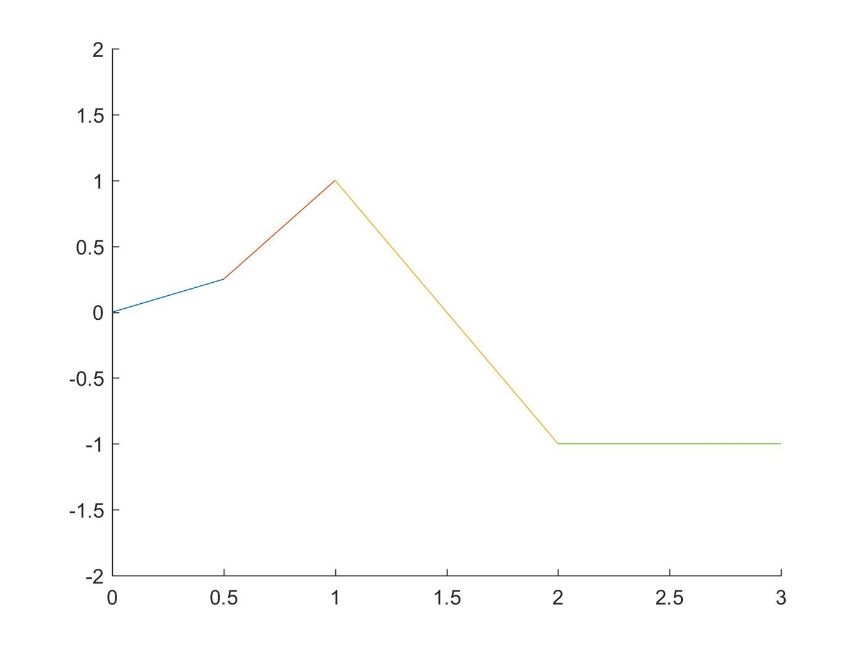
Jared Rivera 26 April 2017 804603106

CEE 103 HW#3 Scripts

%HW3 P1a

clear all; close all; clc;

x1=linspace(0,0.5,100);

x2=linspace(0.5,1,100);

x3=linspace(1,2,100);

x4=linspace(2,3,100);

L1=0.5\*x1;

L2=1.5\*x2-0.5;

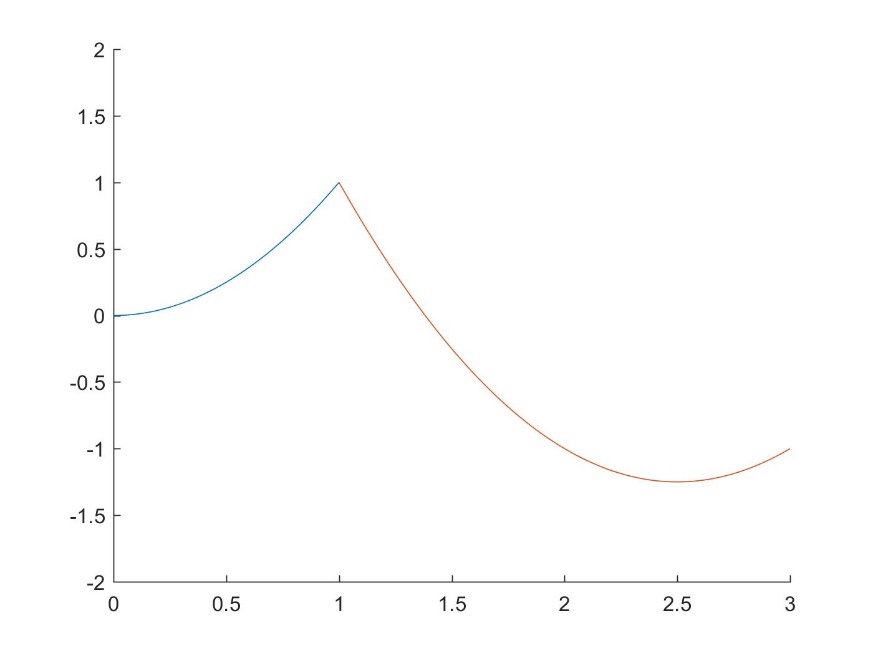
L3=-2\*x3+3;

L4=zeros(100)-1;

hold on

plot(x1,L1,x2,L2,x3,L3,x4,L4);

axis([0,3,-2,2]);



%HW3 P1b

clear all; close all; clc;

x1=linspace(0,1,100);

x2=linspace(1,3,100);

Q1=x1.^2;

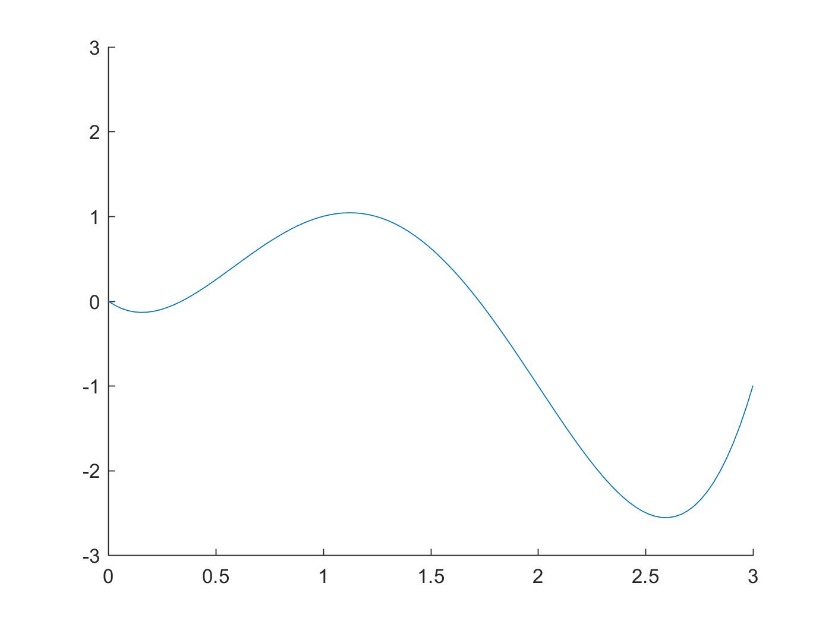
Q2=x2.^2-5\*x2+5;

hold on

plot(x1,Q1,x2,Q2);

axis([0,3,-2,2]);

%HW3 P1c

clear all; close all; clc;

x=linspace(0,3,100);

P1=0.5\*x;

P2=P1+x.\*(x-0.5);

P3=P2-(5/3)\*x.\*(x-0.5).\*(x-1);

P4=P3+1\*x.\*(x-0.5).\*(x-1).\*(x-2);

hold on

plot(x,P4);

axis([0,3,-3,3]);

%HW3 P1d

clear all; close all; clc;

x1=linspace(0,0.5,100);

x2=linspace(0.5,1,100);

x3=linspace(1,2,100);

x4=linspace(2,3,100);

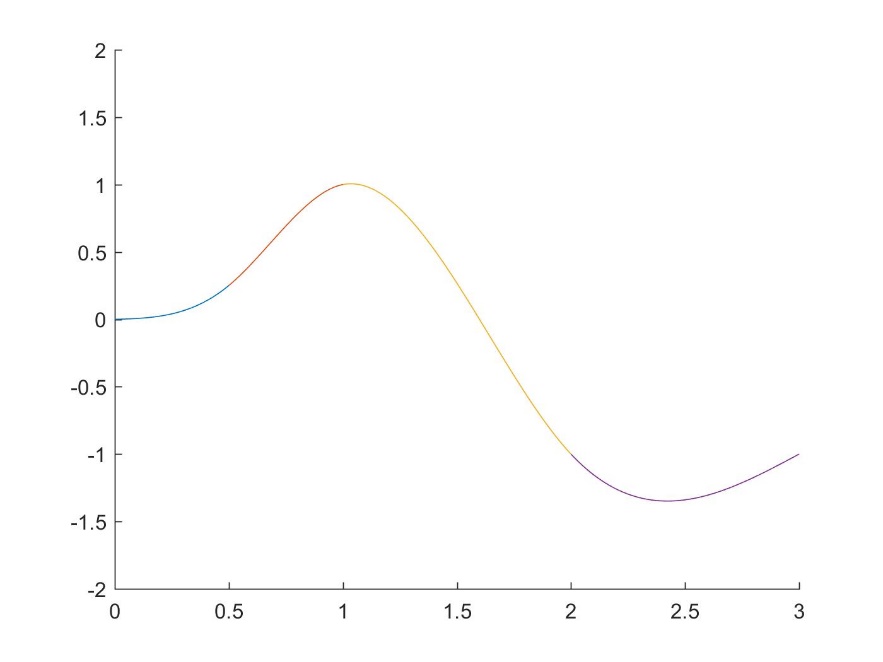
s0=1.81\*x1.^3+0.05\*x1;

s1=1.81\*(1-x2).^3-3.238\*(x2-0.5).^3+2.76\*x2-1.355;

s2=-1.62\*(2-x3).^3+0.905\*(x3-1).^3-4.525\*x3+7.145;

s3=0.905\*(3-x4).^3+0.905\*x4-3.715;

hold on

plot(x1,s0,x2,s1,x3,s2,x4,s3);

axis([0,3,-2,2]);

%HW3 P3

clear all; close all; clc;

x1=linspace(0,1,100);

x2=linspace(1,2,100);

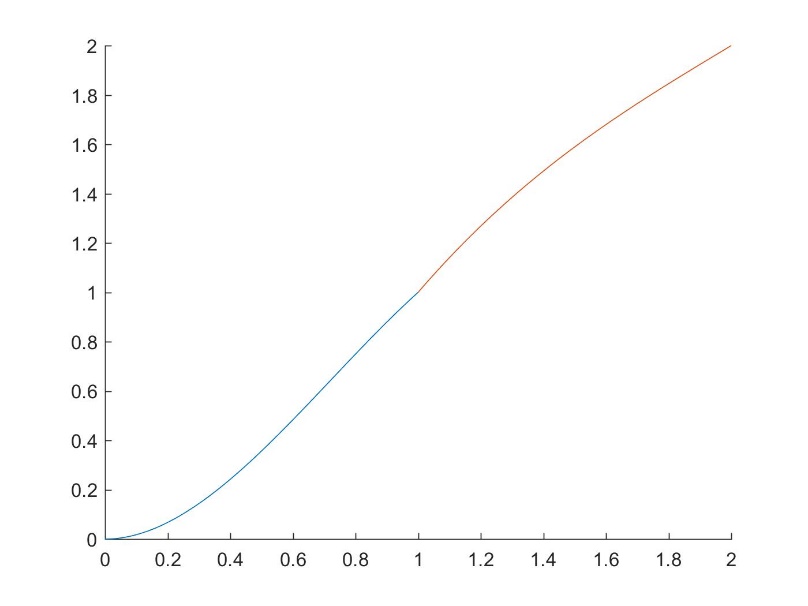
s0=0.619\*(1-x1).^3-0.238\*x1.^3+1.857\*x1-0.619;

s1=-0.238\*(2-x2).^3+0.762\*x2+0.476;

hold on

plot(x1,s0,x2,s1);

axis([0,2,0,2]);



%HW3 P4

clear all; close all; clc;

%Initialize the arrays holding displacement info

x=[0,0.3,0.5,0.8,1,1.2,1.6,1.9,2.1,2.3,2.5];

u=[0,0.2141,0.2938,0.3258,0.3,0.2506,0.1498,0.1613,0.2627,0.4777,0.8438];

EpsilExact=(1/10)\*(4\*x.^3-3\*x.^2-12\*x+9);

%Determine the Lagrange coefficients

for i=1:10

m(i)=(u(i)-u(i+1))/(x(i)-x(i+1));

b(i)=(u(i)\*x(i+1)-u(i+1)\*x(i))/(x(i+1)-x(i));

end

%Determine the Lagrange functions

p1=m(1)\*x+b(1); %on [x1,x2]

p2=m(2)\*x+b(2); %on [x2,x3]

p3=m(3)\*x+b(3); %on [x3,x4]

p4=m(4)\*x+b(4); %on [x4,x5]

p5=m(5)\*x+b(5); %on [x5,x6]

p6=m(6)\*x+b(6); %on [x6,x7]

p7=m(7)\*x+b(7); %on [x7,x8]

p8=m(8)\*x+b(8); %on [x8,x9]

p9=m(9)\*x+b(9); %on [x9,x10]

p10=m(10)\*x+b(10); %on [x10,x11]

%Differentiate the Lagrange functions to get the strain function

ei1=m(1); %on [x1,x2]

ei2=m(2); %on [x2,x3]

ei3=m(3); %on [x3,x4]

ei4=m(4); %on [x4,x5]

ei5=m(5); %on [x5,x6]

ei6=m(6); %on [x6,x7]

ei7=m(7); %on [x7,x8]

ei8=m(8); %on [x8,x9]

ei9=m(9); %on [x9,x10]

ei10=m(10); %on [x10,x11]

%Smooth the strain function according to the algorithm outlined

en(1)=ei1;

for i=2:10

en(i)=(m(i)+m(i-1))/2;

end

en(11)=ei10;

%Determine the Lagrange coefficients

for i=1:10

me(i)=(en(i)-en(i+1))/(x(i)-x(i+1));

be(i)=(en(i)\*x(i+1)-en(i+1)\*x(i))/(x(i+1)-x(i));

end

%Determine the Lagrange functions

e1=me(1)\*x+be(1); %on [x1,x2]

e2=me(2)\*x+be(2); %on [x2,x3]

e3=me(3)\*x+be(3); %on [x3,x4]

e4=me(4)\*x+be(4); %on [x4,x5]

e5=me(5)\*x+be(5); %on [x5,x6]

e6=me(6)\*x+be(6); %on [x6,x7]

e7=me(7)\*x+be(7); %on [x7,x8]

e8=me(8)\*x+be(8); %on [x8,x9]

e9=me(9)\*x+be(9); %on [x9,x10]

e10=me(10)\*x+be(10); %on [x10,x11]

%Determine Error in strain

E1=EpsilExact-e1;

E2=EpsilExact-e2;

E3=EpsilExact-e3;

E4=EpsilExact-e4;

E5=EpsilExact-e5;

E6=EpsilExact-e6;

E7=EpsilExact-e7;

E8=EpsilExact-e8;

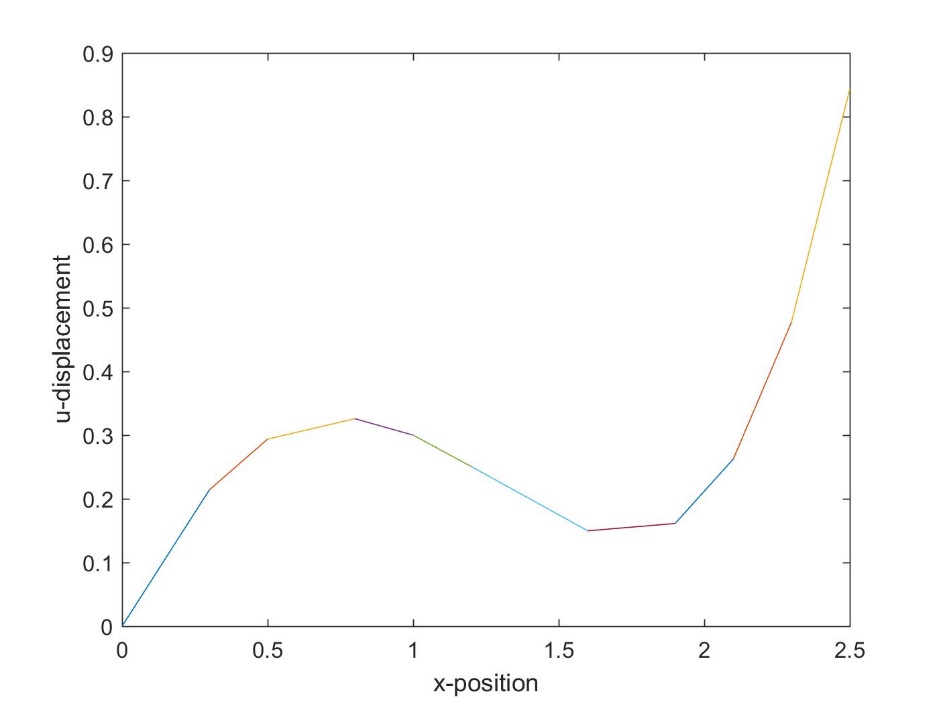
E9=EpsilExact-e9;

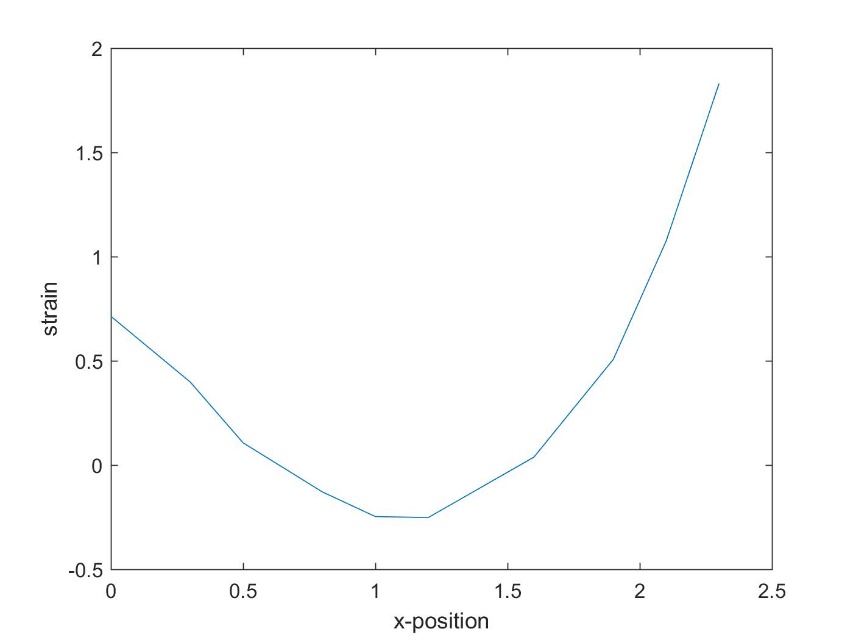
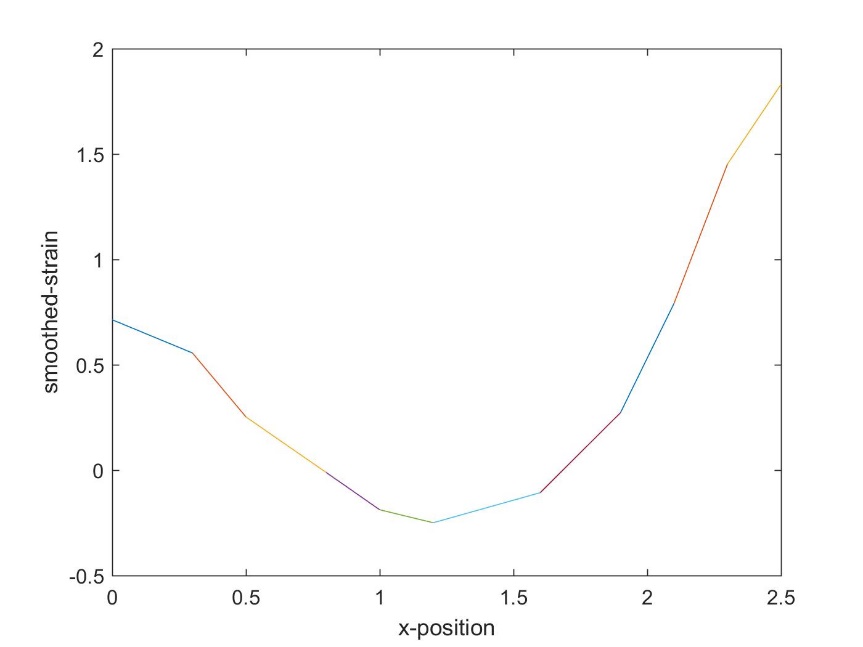
E10=EpsilExact-e10;

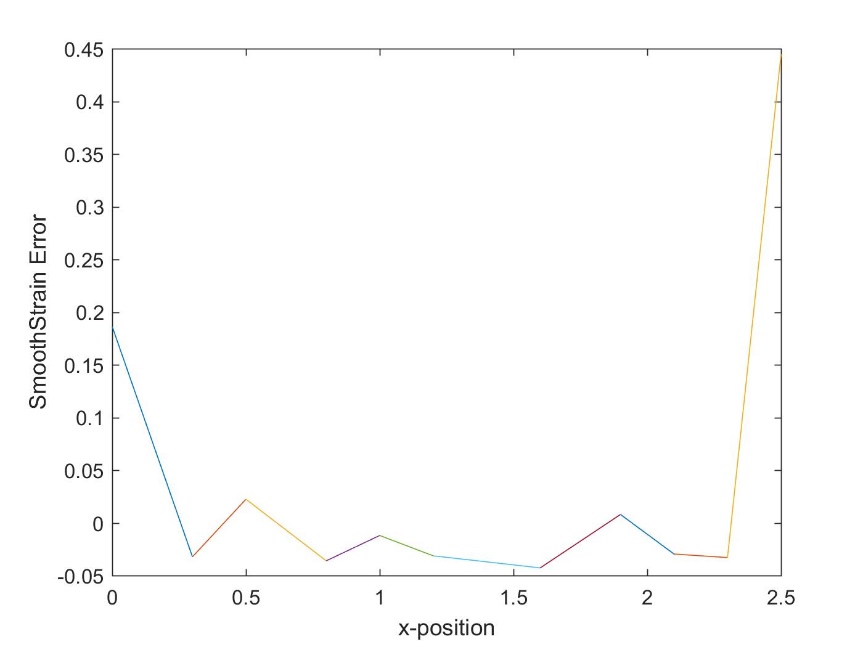
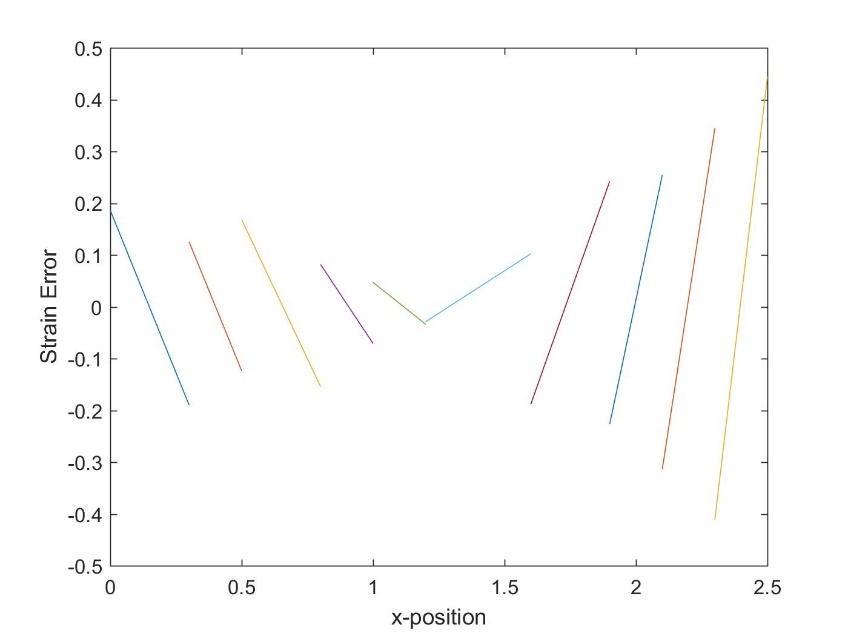
plot(x(1:2),E1(1:2),x(2:3),E2(2:3),x(3:4),E3(3:4),x(4:5),E4(4:5),x(5:6),E5(5:6),x(6:7),E6(6:7),x(7:8),E7(7:8),x(8:9),E8(8:9),x(9:10),E9(9:10),x(10:11),E10(10:11));

xlabel('x-position');

ylabel('Strain Error');







On Error:

As can be seen in the figures above, the error between strain curves has local maxima at the endpoints. A big difference, though, comes from the fact that the smooth strain curve has continuous, low error, while the original strain curve is non-continuous as increases linearly (by a lot) as the x-position leaves the 1.2 region. The average error is much lower on the smooth strain curve due to the fact that by using a technique similar to a spline the curve fits much better with the exact curve.